

Informatica Scenario

Authorize Payment-Part-2 Authorize Payment API

Informatica checks for **Current_Stage** column and **PAYMENT_INTENT_ID_REQUEST** column.

1) a) If **Current_Stage:Auth_Wait** and **PAYMENT_INTENT_ID_REQUEST** is **populated**, Informatica will make **Authorization REST API** call to capture the transaction. Then updates **Current_Stage:Auth_RET** and **PAYMENT_INTENT_ID_RESPONSE** in OMS table. Considered **TRANSACTION_RETURN_CODE: Success(200)**.
b) When **TRANSACTION_RETURN_CODE:Not Success(other than 200)**. Then updates **EXT_PP_RESPONSE**(limit character length to 4000) with response value from TIBCO. Also **Current_Stage:Auth_WAIT** and **PAYMENT_INTENT_ID_RESPONSE** as.....in OMS table.

OMS_Input(to INFA)

XXM_PAYMENT_TRANSACTIONS_ID
MERCHANT_ORG_CODE
ORDER_NUMBER
TRANSACTION_REFERENCE_ID
Current-Stage-AUTH-WAIT
EXT_PP_CUSTOMER_ID
PAYMENT_METHOD_ID
TRANSACTION_AMOUNT
CREATION_DATE
CREATED_BY
METADATA columns as needed by source systems

Authorize Payment-Part-1 Create Payment API

2) a) If **Current_Stage:Auth_Wait** and **PAYMENT_INTENT_ID_REQUEST** is **not populated**, Informatica will make call to create a new **paymentIntentID(Create_Payment API Called)**, Then updates **Current_Stage:Auth_Wait** and **PAYMENT_INTENT_ID_RESPONSE** in **OMS** table. Considered **TRANSACTION_RETURN_CODE:Success(200)**. Next **Authorize it (Authorize_Payment API)** Part 1a(above).

b) When **TRANSACTION_RETURN_CODE:Not Success**(other than 200). Then updates **EXT_PP_RESPONSE**(limit character length to 4000) with response value from **TIBCO**.Also updates **Current_Stage: Auth_Wait** and **PAYMENT_INTENT_ID_RESPONSE** as..... in **OMS** table. Call **Create Payment API** Again to create **payment Intent ID** and then **Authorize**.

Capture Payment

Capture Payment Part-1 Using Capture Rest API Informatica batch settlement (authorized payments)

1) a) If **Current_Stage:Capture_Wait** and **PAYMENT_INTENT_ID_REQUEST** is **populated**, Informatica will make **Capture REST API** call to capture the transaction.Then updates **Current_Stage:Capture_RET** and **PAYMENT_INTENT_ID_RESPONSE** in **OMS** table. Considered **TRANSACTION_RETURN_CODE:Success(200)**.

b) When **TRANSACTION_RETURN_CODE:Not Success**(other than 200). Then updates **EXT_PP_RESPONSE**(limit character length to 4000) with response value from **TIBCO**.Also **Current_Stage:Capture_WAIT** and **PAYMENT_INTENT_ID_RESPONSE** as in **OMS** table. **Capture API** is called again.

XXM_PAYMENT_TRANSACTIONS_ID **PAYMENT_INTENT_ID_REQUEST**
MERCHANT_ORG_CODE
ORDER_NUMBER
TRANSACTION_REFERENCE_ID
Current-Stage-AUTH-WAIT
EXT_PP_CUSTOMER_ID
PAYMENT_METHOD_ID
TRANSACTION_AMOUNT
CREATION_DATE
CREATED_BY
METADATA columns as needed by source systems

XXM_PAYMENT_TRANSACTIONS_ID**MERCHANT_ORG_CODE****ORDER_NUMBER****TRANSACTION_REFERENCE_ID**
Current-Stage-CAPTURE-WAIT
EXT_PP_CUSTOMER_ID**PAYMENT_METHOD_ID****PAYMENT_INTENT_ID_REQUEST: V**
paymentIntentID that has already been captured or has expired.
TRANSACTION_AMOUNT**CREATION_DATE****CREATED_BY****METADATA** columns as needed by source systems

Capture Payment Part-2 Using Create Rest API
2) Informatica batch settlement (not authorized payments)
If PAYMENT_INTENT_ID_REQUEST is **not populated**, Informatica will make call to create a **new paymentIntentID**, authorize and then capture it.
Current_Stage-CAPTURE-WAIT is Updated to **AUTH-WAIT** and Authorization process is followed.

XXM_PAYMENT_TRANSACTIONS_ID
MERCHANT_ORG_CODE
ORDER_NUMBER
TRANSACTION_REFERENCE_ID
Current-Stage-CAPTURE-WAIT
EXT_PP_CUSTOMER_ID
PAYMENT_METHOD_ID
PAYMENT_INTENT_ID_REQUEST: Will be blank for paymentIntentID that has already been expired.
TRANSACTION_AMOUNT
CREATION_DATE
CREATED_BY
METADATA columns as needed by source systems

Refund Payment
Refund Payment
Informatica checks for **Current_Stage** column and **PAYMENT_INTENT_ID_REQUEST** column1) a) If **Current_Stage:Refund_Wait** and **PAYMENT_INTENT_ID_REQUEST** must be **populated**, Informatica will make **Capture REST API call** to capture the transaction. Then updates **Current_Stage:Refund_RET** and **PAYMENT_INTENT_ID_RESPONSE** in **OMS table**. Considered **TRANSACTION_RETURN_CODE:Success(200)**.
b) When **TRANSACTION_RETURN_CODE:Not Success**(other than 200). Then updates **EXT_PP_RESPONSE**(limit character length to 4000) with response value from **TIBCO**. Also **Current_Stage:Capture_WAIT** and **PAYMENT_INTENT_ID_RESPONSE** as..... in **OMS table**. Call **Capture Rest API** again.

XXM_PAYMENT_TRANSACTIONS_IDMERCHANT_ORG_CODEORDER_NUMBERTRANSACTION_REFERENCE_ID
Current-Stage-REFUND-WAIT
EXT_PP_CUSTOMER_IDPAYMENT_METHOD_IDPAYMENT_INTENT_ID_REQUEST:MERCHANT_ORG_CODEORDER_NUMBERTRANSACTION_REFERENCE_ID
refundTRANSACTION_AMOUNTCREATION_DATECREATED_BYMETADATA columns as needed by source systems

PAYMENT_INTENT_ID_RESPONSE: Should always be populated as a part of response.

CURRENT_STAGE-**AUTH-RET**

TRANSACTION_RETURN_CODE

TRANSACTION_DATE

LAST_UPDATED_DATE

LAST_UPDATED_BY

DATE_PROCESSED

EXT_PP_RESPONSE(In case of non success response)

PAYMENT_INTENT_ID_RESPONSE: Should always be populated as a part of response.

CURRENT_STAGE-**AUTH-WAIT**

TRANSACTION_RETURN_CODE

TRANSACTION_DATE

LAST_UPDATED_DATE

LAST_UPDATED_BY

DATE_PROCESSED

EXT_PP_RESPONSE(In case of non success response)

PAYMENT_INTENT_ID_RESPONSE-Should always be populated as a part of response.CURRENT_STAGE-**CAPTURE-RET**TRANSACTION_RETURN_CODETRANSACTION_DATELAST_UPDATED_DATELAST_UPDATED_BYDATE_PROCESSEDEXT_PP_RESPONSE

PAYMENT_INTENT_ID_RESPONSE: Should always be populated as a part of response.
CURRENT_STAGE-**AUTH-WAIT**
TRANSACTION_RETURN_CODE
TRANSACTION_DATE
LAST_UPDATED_DATE
LAST_UPDATED_BY
DATE_PROCESSED
EXT_PP_RESPONSE(In case of non success response)

PAYMENT_INTENT_ID_RESPONSE: Should always be populated as a part of response.CURRENT_STAGE-**REFUND-RET**TRANSACTION_RETURN_CODETRANSACTION_DATELAST_UPDATED_DATELAST_UPDATED_BYDATE_PROCESSEDEXT_PP_RESPONSE